PAMHYR2: A GRAPHICAL USER INTERFACE FOR 1D HYDRO-SEDIMENTARY MODELLING OF RIVERS

Pierre-Antoine Rouby[1], Benoît Camenen, Lionel Pénard, Léa Kieffer, Théophile Terraz[1]
INRAE, UR RiverLy, centre de Lyon-Grenoble, 5 rue de la Doua, CS20244, 69625, Villeurbanne, France.
pierre-antoine.rouby@inrae.fr, benoit.camenen@inrae.fr, lionel.penard@inrae.fr,
lea.kieffer@inrae.fr, theophile.terraz@inrae.fr

KEY WORDS

1D modelling, sedimentary modelling, user interface

ABSTRACT

In numerical river simulation, one-dimensional models are commonly used to study water level and discharge for large domains or long time series. These models are less time-consuming than two- and three-dimensional numerical models, and require fewer input parameters and allow ensemble runs. To build a one-dimensional hydraulic model, a pre- and post-processing tool is needed for creating reach geometry, specifying initial and boundary conditions, friction coefficients and other numerical parameters. Such a tool needs to ensure the consistency of the model and provide a user-friendly graphical user interface. In this article, we present Pamhyr2, the fully rebuilt version of the PAMHYR modelling platform. It is developed using Python, PyQt and Matplotlib. Pamhyr2 is free and open-source, multilingual, cross-platform (Linux, Windows) and is generic enough to accept various one-dimensional solvers. Pamhyr2 includes and enhances all the features from the previous version: multiple-reach modelling, geometry definition from cross-section data, initial and boundary conditions, friction coefficients, hydraulic structures, lateral inflow, punctual intake and the results visualization window. In addition, this version includes new features such as pollutants modelling, bed-load and bed evolution. We describe several windows: the creation of sedimentary layers in the river bed, the sediment characteristics for each layer, the sediment, and pollutant boundary conditions and the lateral inputs. These functionalities are illustrated with simple examples. We finally show the visualization windows for bed evolution.

1. INTRODUCTION

Numerous tools are available to simulate river flow. Among them, one-dimensional (1D) numerical hydraulic models have the advantage of being accurate while being inexpensive in terms of computation time, making it possible to simulate long rivers over very long timescales. A lot of 1D numerical solvers for river hydraulics are available, including HEC-RAS[2], Mike 11[3] and Mascaret[4] for example. However, in order to investigate specific research issues it is essential to have the ability to edit the source code of numeric solvers and to have an ergonomic Graphical User Interace (GUI). Most of the widespread 1D modeling tools are either written with a proprietary source code (such as HEC-RAS and Mike 11) or are research tools with irregularly maintained GUI. On its side, INRAE has been developing two 1D solvers for over 40 years: Mage[5] [5] and RubarBE[6]. In parallel, a GUI named PAMHyR has been developed to enable modelers to build river models, parameterize calculations, launch Mage and RubarBE solvers and post-process the results [3].

PAMHyR, developed in Java since the end of 90s (originally named X-NT), it is written with the following constraints in mind:

- Multi-platform Windows, Unix-like,

[1] Corresponding author
[2] HEC-RAS: https://www.hec.usace.army.mil/software/hec-ras
[3] Mike11: https://www.mikepoweredbydhi.com/products/mike-11
[4] Mascaret: http://www.opentelemac.org
[5] Mage: https://gitlab.irstea.fr/jean-baptiste.faure/mage
[6] RubarBE: https://gitlab.irstea.fr/theophile.terraz/rubarbe

- no dependence on an external graphics library with a paid license for user,
- generate as much code as possible from a UML diagram.

Now, PAMHyR suffers from complexity due to highly specific interface, as well as a significant technological debt [4]: old external library no longer maintained, over dated project architecture, and numerous modifications by multiple developers (and internship) over short periods of time, with no vision of long-term maintainability. What's more, the software is only in French and the code is closed (the sources are not available), which is a hindrance to the notoriety of the software as well as to the vitality of development. In addition, no significant changes have been made in the application since 2016, even if PAMHyR is still often used for river modeling [2].

INRAE needs a new tool to keep up with today's technologies and with the development of its numerical solvers. In this paper we introduce the current work on the new graphical user interface Pamhyr2.

2. OBJECTIVE

2.1 History and need

PAMHyR is available for two solvers, with two different types of studies: Mage and RubarBE. These two different kinds of studies have mostly similar interfaces, but have their own specific features. They can both be used to: define reach geometry, boundary conditions, lateral inflows, overflows, friction, and structures. In addition, you can run the solver on the study you are editing, and view the solver's results. Firstly, although both Mage and RubarBE are multiple-reach compatible, only the Mage interface enables multiple-reach study editing. Secondly, many small details change from one interface to the next. In addition, the latest Mage features, for example, are not available for modeling in the interface.

A total overhaul of the Java code would be too much work, given the technological debt accumulated since the beginning of its development. Given the workload involved in modernizing PAMHyR, the decision is made to develop a new version of the software from scratch. This version should be equivalent to the Java version in terms of functionality, and add new features, notably the new Mage functionalities. The aim is also to have an interface that is more modern, more user-friendly, more flexible, available on multiple platforms and in different languages. The interface must also have a certain degree of generality and be able to adapt to different solvers without major changes to the interface. It is also important that the code is open to contributions, to keep the project as lively as possible.

2.2 Technical choices

From a technical point of view, choices were made on the basis of the above-mentioned objectives (modernity, maintainability, multiplatform, multilingual, user-friendliness, flexibility), as well as: the ease of learning the technology to facilitate contribution to the software, the popularity of the technology in the scientific community and the interoperability of this technology.

We chose Python[7] as our programming language. It is a popular language in the scientific world, but also in many other fields, with lot of very active software libraries. Moreover, it is a simple and interpreted language, often recommended for beginners, and practiced by beginners and experienced developers. In addition, there is the PyQT[8] software library that lets you use the Qt[9] GUI library, which is ideal for this project. It is very popular and active, and available on many platforms (GNU/Linux, Windows, MacOS, Unix, BSD, Android, iOS). Qt also includes a number of tools for translating interfaces into different languages. Here we use the community version of Qt, with no paid features. For graphics and visualization, the choice fell on Matplotlib[10], which is very popular in the scientific community and can be used to draw a wide variety of graphics. For saving studies, the choice was made to use SQLite[11], unlike the previous version which used a text file formatted in XML. The advantages of a SQLite database, in addition to being supplied with Python, include ease of use, interoperability, fast, formal structuring by table and consistency

[7] Python: https://www.python.org/
[8] PyQt: https://pypi.org/project/PyQt5/
[9] Qt: https://www.qt.io/
[10] Matplotlib: https://matplotlib.org/
[11] SQLite: https://www.sqlite.org/index.html

checking between tables. We have chosen to version the database to allow it to evolve with the software, and to make it possible to add columns to tables or make other minor modifications to the database without breaking the compatibility of studies from previous versions. However, PAMHyR and Pamhyr2 study files will not be compatible anymore, and previous studies have to be recreated for this version.

This new version Pamhyr2 remains closely linked to the Mage solver with which it is supplied. But the ambition of flexibility and generalization gives us the opportunity to develop an interface that can be adapted to other solvers. There can be two solutions for using a solver with Pamhyr2: either the solver adapts to Pamhyr2, or Pamhyr2 adapts to the solver. So Pamhyr2 must integrate a generic output containing the study data that can be used by other solvers, which will be launched via an external executable, and then a generic result input that must be supplied by the solver. In addition, Pamhyr2 must be able to handle the execution and reading of results for specific solvers, and the addition of a solver to Pamhyr2 must be reasonably easy for the solver developer. We have opted for the free and open source software (FOSS) [6] license GPLv3[12], which guarantees freedom to access, study, modify, redistribute and share. In addition, it guarantees that Pamhyr2's code will remain free and is not allowed to be added any closed-source modifications. However, it will still be possible to use closed-source solvers, because solver is not a part of Pamhyr2.

3. DEVELOPMENTS

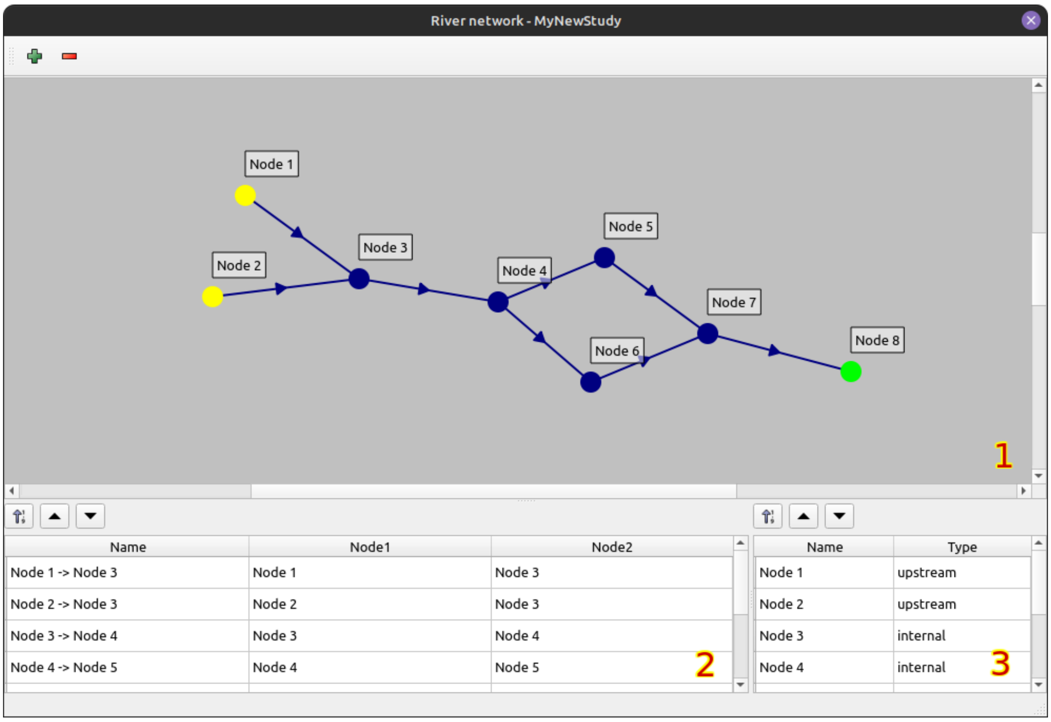
3.1 New version of PAMHYR

Pamhyr2 currently features the following interfaces: Editing of river network, geometry, boundary conditions, lateral contribution, friction, initial conditions, solver numerical parameters, as well as the interface for launching a solver and displaying hydraulic and sediment results.

3.1.1 River network

The river network corresponds to the topology of the hydraulic network of the river. River network editing (**Figure 1**) is performed via a window featuring a graphical network area and two tables, one with a list of nodes (zone 3) and the other with a list of reaches (zone 2). The graphic area (zone 1) lets you add, delete and move nodes. It can also be used to link nodes in the network by arrows that represent reaches. The color of the node in the graphic area depends on whether it is a downstream, upstream or internal network node. If there is more than one reach, you can use this interface to select a reach, which will then be selected for editing in the following reach-dependent windows. It is also possible to deactivate reaches, in which case they will not be used for the solver run. Tables can be used to edit additional information on nodes and reaches, such as giving them names.

[12] GPLv3: https://www.gnu.org/licenses/gpl-3.0.en.html



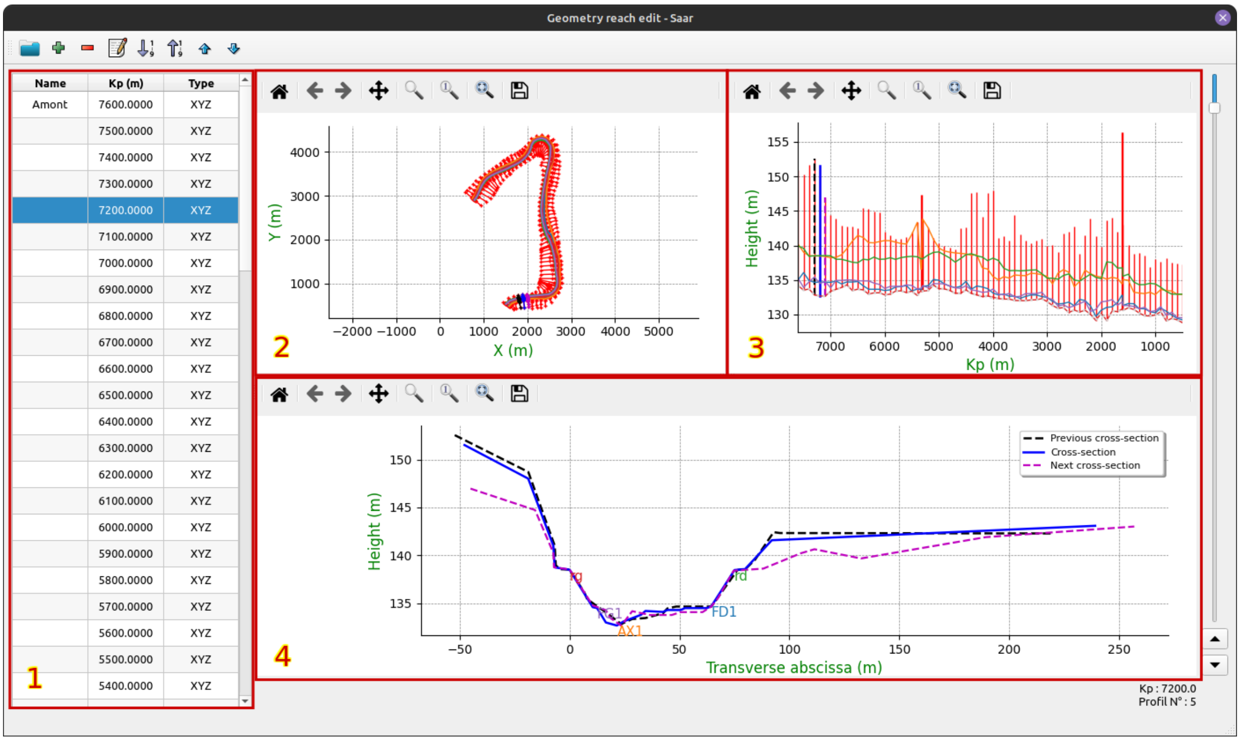
Name	Node1	Node2
Node 1 -> Node 3	Node 1	Node 3
Node 2 -> Node 3	Node 2	Node 3
Node 3 -> Node 4	Node 3	Node 4
Node 4 -> Node 5	Node 4	Node 5

Name	Type
Node 1	upstream
Node 2	upstream
Node 3	internal
Node 4	internal

Figure 1: River network window for a dummy river with 8 reaches, 2 upstream node (yellow), 1 downstream node (green) and 5 internal node (blue).

3.1.2 Geometry

Geometry editing (**Figure 2**) is linked to a reach; the reach used will be the last reach selected in the river network window. This window contains a table with the list of the reach's cross-sections (zone 1) and three graphics. A top view based on the XY positions of the cross-sections' points and guidelines (zone 2). A longitudinal view of all cross-sections with the height of the guidelines' points for each profile (zone 3). Finally, a cross-sectional view is displayed (zone 4) showing the selected cross-section with the previous and the following one in dotted line. Each cross-section is defined by a Kilometer Point (KP) and an optional name. The KP represents the longitudinal coordinate of the cross-section in the numerical solver. It is possible to add cross-sections and then select them to access an editing window, or to import cross-sections from file. Cross-section editing window takes the form of a table showing the raw data of the points that make up the cross-section, and a graph showing the current cross-section.

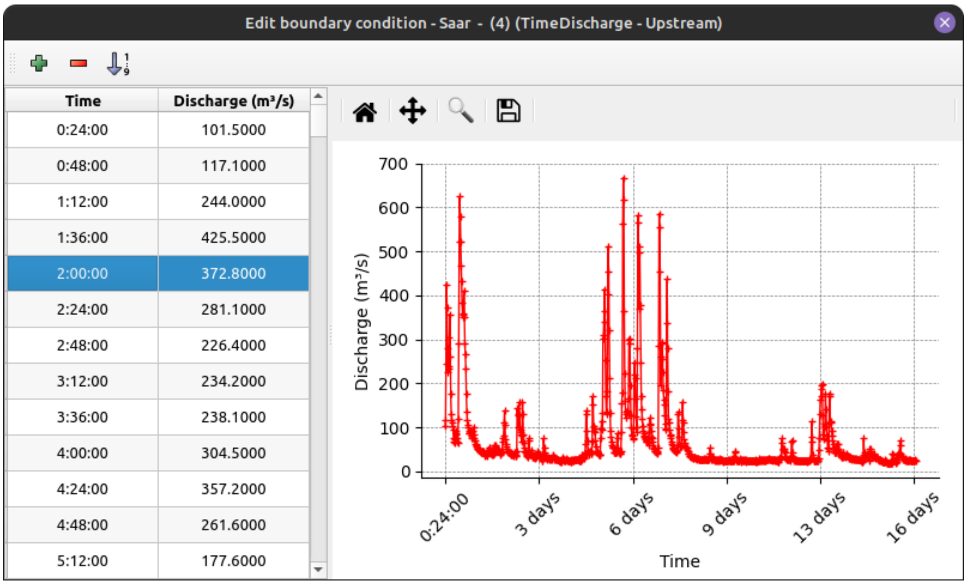


Name	Kp (m)	Type
Amont	7600.0000	XYZ
	7500.0000	XYZ
	7400.0000	XYZ
	7300.0000	XYZ
	7200.0000	XYZ
	7100.0000	XYZ
	7000.0000	XYZ
	6900.0000	XYZ
	6800.0000	XYZ
	6700.0000	XYZ
	6600.0000	XYZ
	6500.0000	XYZ
	6400.0000	XYZ
	6300.0000	XYZ
	6200.0000	XYZ
	6100.0000	XYZ
	6000.0000	XYZ
	5900.0000	XYZ
	5800.0000	XYZ
	5700.0000	XYZ
	5600.0000	XYZ
	5500.0000	XYZ
	5400.0000	XYZ

Figure 2: Geometry window for Saar river on the unique reach of the Saar river study.

3.1.3 Boundary conditions

Boundary conditions (BC) are associated with a network node. There are three different types: hydrograph, water level chart or rating curve, in three different categories: Liquid, solid and suspension. Boundary conditions can be added, deleted and edited. Editing a BC (**Figure 3**) takes the form of a two-column table (the type of column varies according to the type of BC) and a graph representing the data. Data can be copied and pasted from a spreadsheet, or edited by hand.



Time	Discharge (m³/s)
0:24:00	101.5000
0:48:00	117.1000
1:12:00	244.0000
1:36:00	425.5000
2:00:00	372.8000
2:24:00	281.1000
2:48:00	226.4000
3:12:00	234.2000
3:36:00	238.1000
4:00:00	304.5000
4:24:00	357.2000
4:48:00	261.6000
5:12:00	177.6000

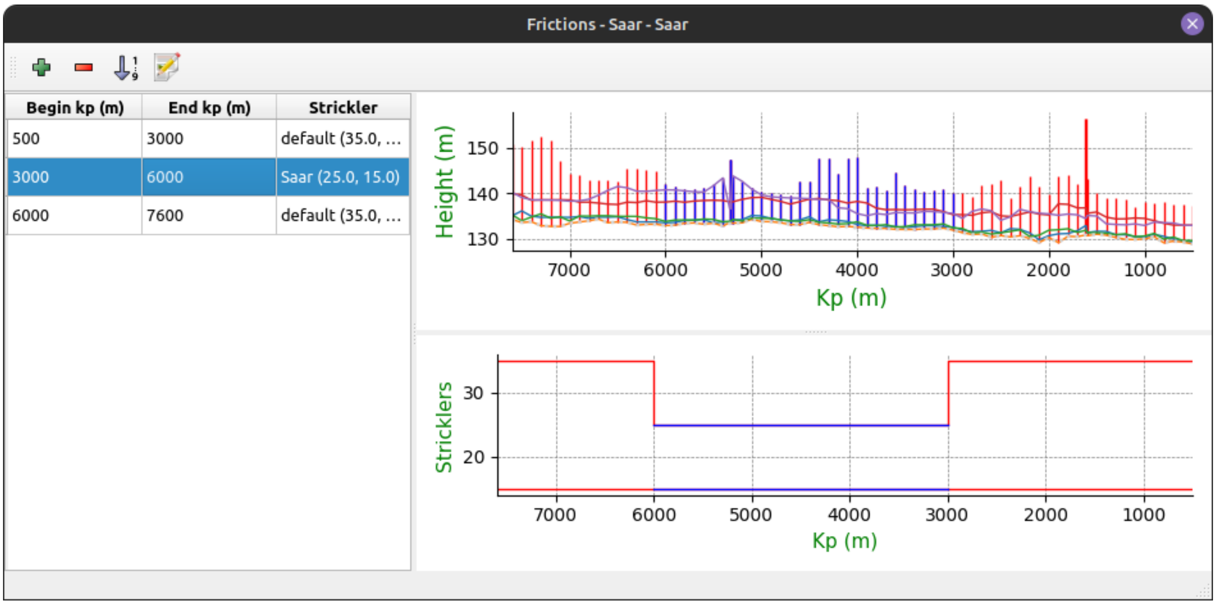
Figure 3: Edition window of the boundary conditions hydrograph for the Saar river upstream node.

3.1.3 Lateral contribution

The user interface for lateral contributions is similar to that for boundary conditions, except that it applies to a part of a reach between two KP, and not to a node.

3.1.4 Friction

Friction editing is linked to a reach, a friction coefficient is defined and then applied to a portion of the reach between two KP (**Figure 4**). Friction is defined by two Strickler coefficients, one for the river's main channel and one for the flood plain. The list of available Strickler coefficients is defined at the study level.

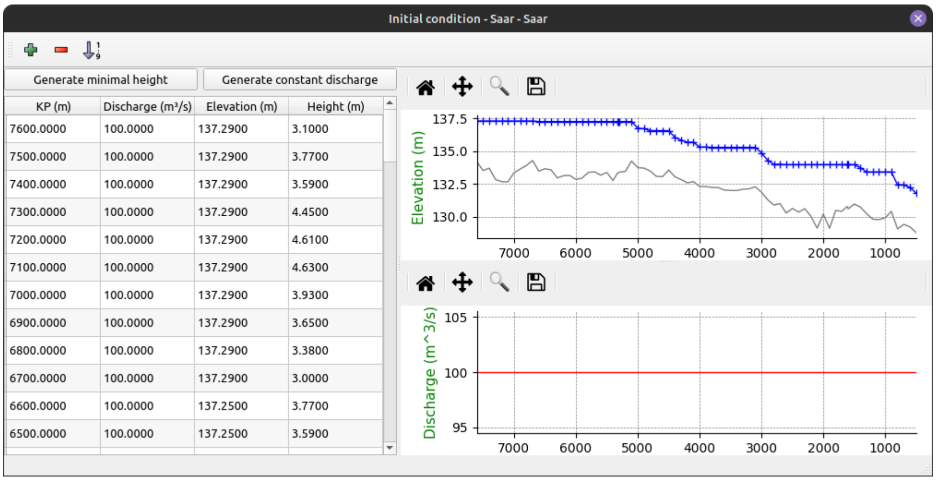


Begin kp (m)	End kp (m)	Strickler
500	3000	default (35.0, ...
3000	6000	Saar (25.0, 15.0)
6000	7600	default (35.0, ...

Figure 4: The reach portions for friction coefficient edition window for the Saar study.

3.1.5 Initial conditions

The initial conditions (**Figure 5**) are also linked to the reach, corresponding to the water level and flow rate at the initial time of the simulation. These two values can be defined separately for each cross-section of the reach. It is also possible to use predefined functions to generate a water line and flow rate.

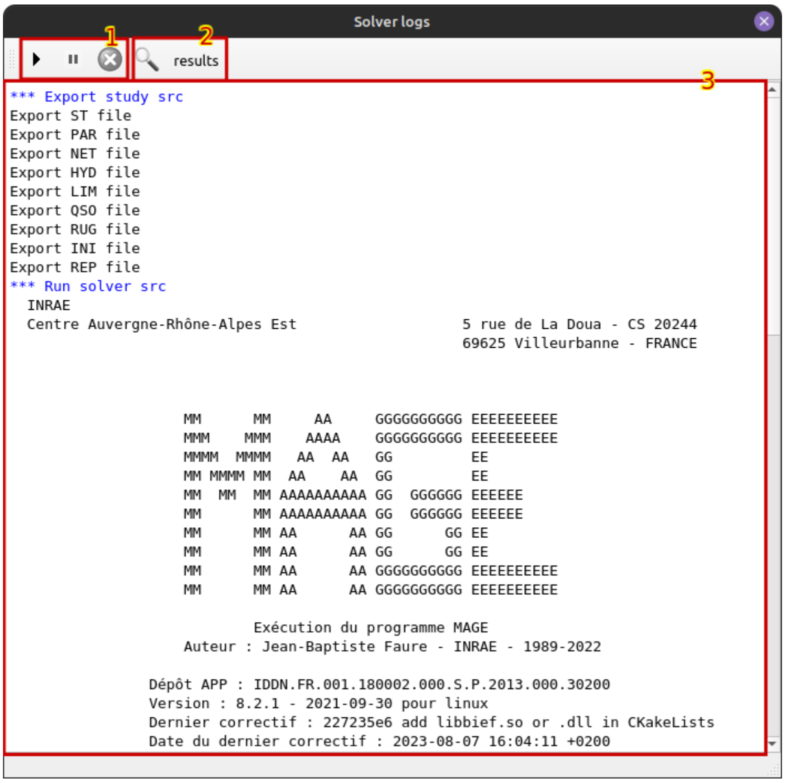


KP (m)	Discharge (m³/s)	Elevation (m)	Height (m)
7600.0000	100.0000	137.2900	3.1000
7500.0000	100.0000	137.2900	3.7700
7400.0000	100.0000	137.2900	3.5900
7300.0000	100.0000	137.2900	4.4500
7200.0000	100.0000	137.2900	4.6100
7100.0000	100.0000	137.2900	4.6300
7000.0000	100.0000	137.2900	3.9300
6900.0000	100.0000	137.2900	3.6500
6800.0000	100.0000	137.2900	3.3800
6700.0000	100.0000	137.2900	3.0000
6600.0000	100.0000	137.2500	3.7700
6500.0000	100.0000	137.2500	3.5900

Figure 5: Initial conditions edition window for the Saar study and the Saar reach. Initial conditions initialized with minimal height of 3 meter and a constant flow rate of 100 m^3/s.

3.1.6 Running a solver

Solver numerical parameters are defined using a key/value table specific to each type of solver known to Pamhyr2. You can select and run the desired solver from the list of configured solvers. Then a window appears with a text area where you can follow the progress of file export and solver output in **Figure 6** (zone 3). You can also pause, stop or restart the solver using buttons (zone 1). Once the simulation has finished, you can open a solver trace file, and if the simulation has finished correctly, you can open the results using the "results" button (zone 2).

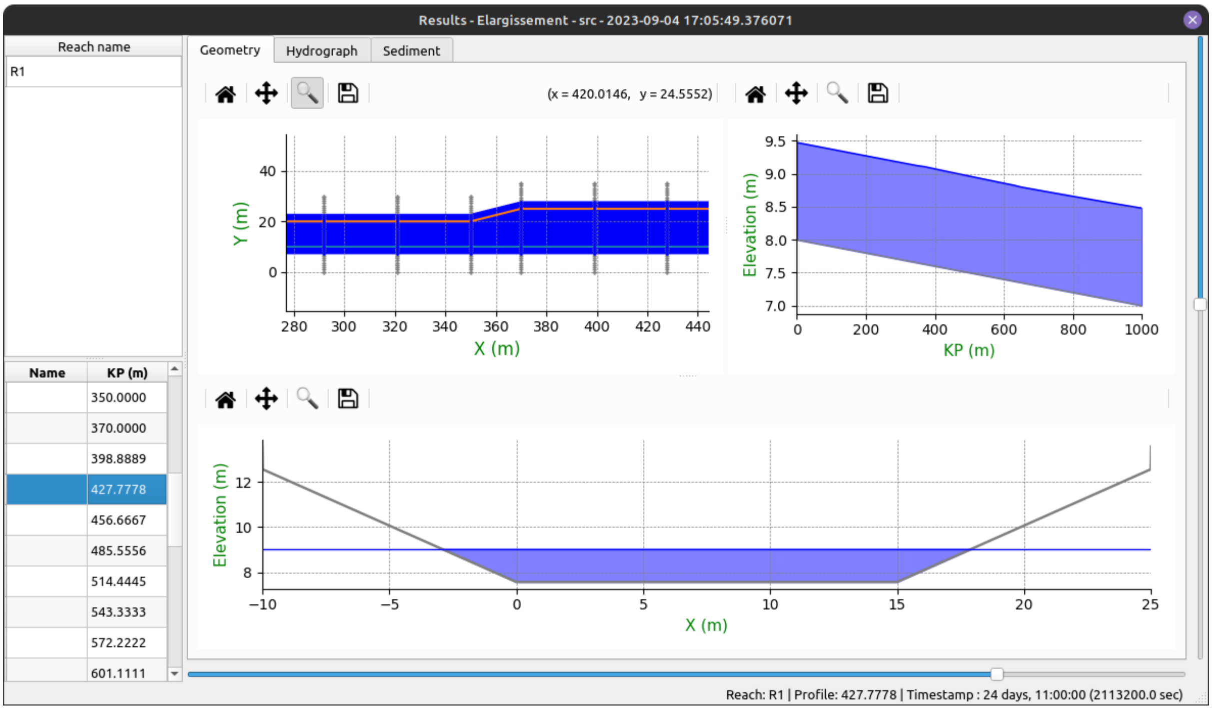


```
*** Export study src
Export ST file
Export PAR file
Export NET file
Export HYD file
Export LIM file
Export QSO file
Export RUG file
Export INI file
Export REP file
*** Run solver src
  INRAE
  Centre Auvergne-Rhône-Alpes Est                   5 rue de La Doua - CS 20244
                                                    69625 Villeurbanne - FRANCE



                    MM      MM     AA     GGGGGGGGGG EEEEEEEEEE
                    MMM    MMM    AAAA    GGGGGGGGGG EEEEEEEEEE
                    MMMM  MMMM   AA  AA   GG         EE
                    MM MMMM MM  AA    AA  GG         EE
                    MM  MM  MM AAAAAAAAAA GG  GGGGGG EEEEEE
                    MM      MM AAAAAAAAAA GG  GGGGGG EEEEEE
                    MM      MM AA      AA GG      GG EE
                    MM      MM AA      AA GG      GG EE
                    MM      MM AA      AA GGGGGGGGGG EEEEEEEEEE
                    MM      MM AA      AA GGGGGGGGGG EEEEEEEEEE

                            Exécution du programme MAGE
                    Auteur : Jean-Baptiste Faure - INRAE - 1989-2022

                Dépôt APP : IDDN.FR.001.180002.000.S.P.2013.000.30200
                Version : 8.2.1 - 2021-09-30 pour linux
                Dernier correctif : 227235e6 add libbief.so or .dll in CKakeLists
                Date du dernier correctif : 2023-08-07 16:04:11 +0200
```

Figure 6: Solver log window for "*Elargissement*" tests case with a mage solver named "src".

3.1.7 Results processing and visualization

The post processing window is made up of tables allowing you to select the reaches, profiles, and timestamps for visualizing the results. The results are presented in different tabs according to the type of data displayed. The **Figure 7** shows hydraulic data with water level at a given reach, profile, and time. The graphs shown here are the same as those in the geometry window, with the addition of water level results at a given time.



Results - Elargissement - src - 2023-09-04 17:05:49.376071

Reach name
R1

Name	KP (m)
	350.0000
	370.0000
	398.8889
	427.7778
	456.6667
	485.5556
	514.4445
	543.3333
	572.2222
	601.1111

(x = 420.0146, y = 24.5552)

Reach: R1 | Profile: 427.7778 | Timestamp : 24 days, 11:00:00 (2113200.0 sec)

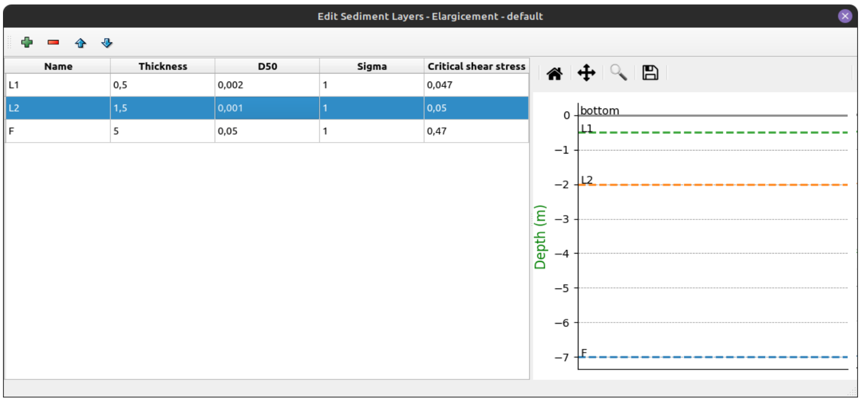
Figure 7: Hydraulics results window for “*Elargissement*” tests case with a Mage solver at timestamps 2113200.0 sec for reach R1 and cross-section at KP 427.7778.

3.2 New features

Some new features are already implemented in Pamhyr2. First of all, the software is multilingual with English and French currently available. An other very useful enhancement, is the possibility to use the “Ctrl+Z” or “Ctrl+Shift+Z” shortcuts to undo or redo the user actions.

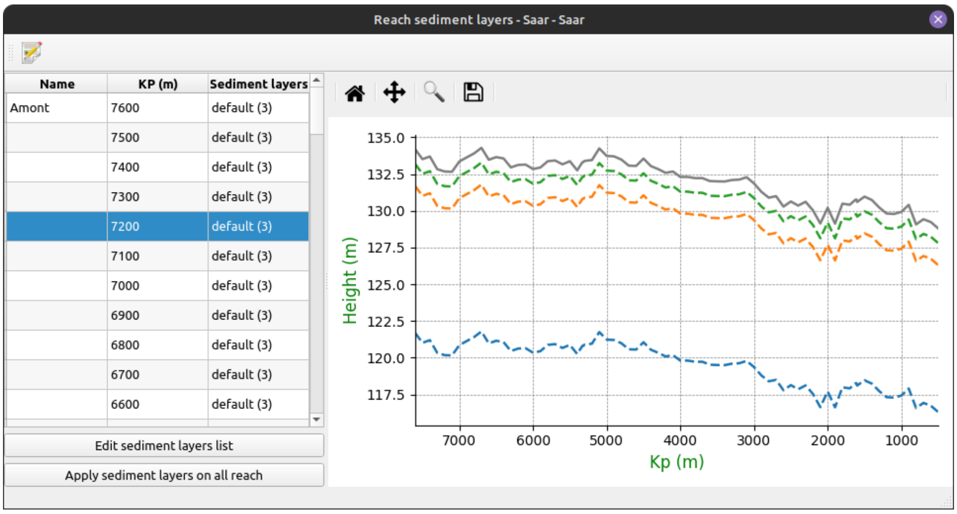
Another novelty lies in the initial conditions window. Previously the water line could be generated by constant height or altitude. It is now possible to give a minimum height with an increasing function from downstream to upstream, thus avoiding physically unrealistic water holes and getting a little closer to a real and physically coherent water line (**Figure 5**).

An important new feature in this version is the presence of sediment layers. They can be defined as shown in **Figure 8.** Sediment layers can be applied separately for each cross-section (**Figure 9**). As with friction, they must first be defined and then applied to a profile. They can also be applied separately to each point of a cross-section. Layers are defined by a thickness, a median diameter (D50), a grain size sorting coefficient (Sigma) and a critical shear stress as described in Pierre Balayn PhD thesis [1]. The visualization of sedimentary results is currently done using a plot of the river profile, which shows the evolution of the bed bottom and the thickness of layers as a function of time (**Figure 10**).



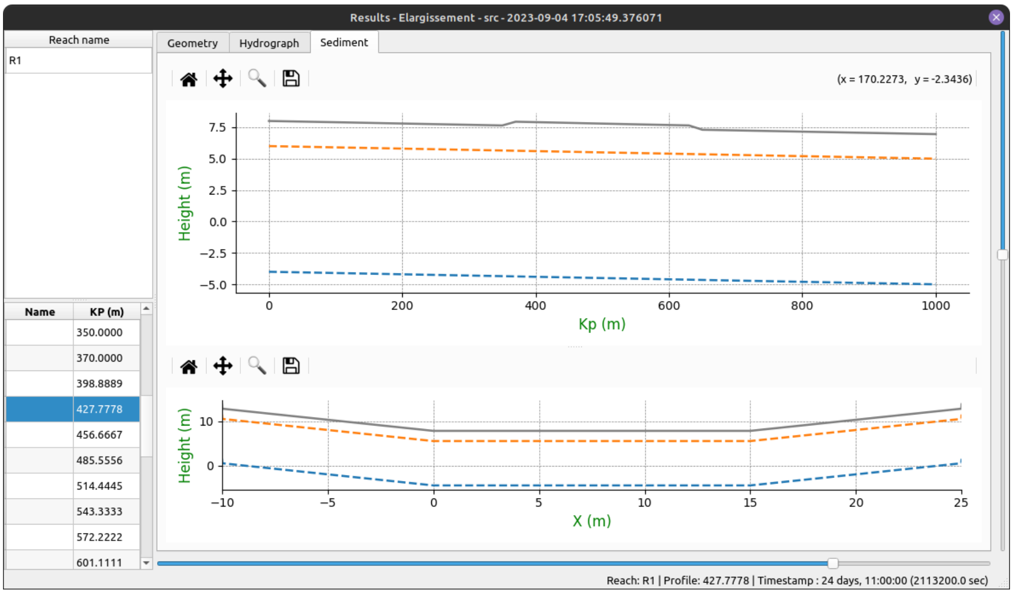
Name	Thickness	D50	Sigma	Critical shear stress
L1	0,5	0,002	1	0,047
L2	1,5	0,001	1	0,05
F	5	0,05	1	0,47

Figure 8: Sediment layers definition window for Saar study. This sediment layers has three layers (L1, L2 and F).


Reach sediment layers - Saar - Saar
Edit sediment layers list
Apply sediment layers on all reach
Height (m)
135.0
132.5
130.0
127.5
125.0
122.5
120.0
117.5
7000
6000
5000
4000
3000
2000
1000
Kp (m)

Name	KP (m)	Sediment layers
Amont	7600	default (3)
	7500	default (3)
	7400	default (3)
	7300	default (3)
	7200	default (3)
	7100	default (3)
	7000	default (3)
	6900	default (3)
	6800	default (3)
	6700	default (3)
	6600	default (3)

Figure 9: Sediment layers defined in **Figure 8** applied to the reach Saar.



Reach name
R1

Name	KP (m)
	350.0000
	370.0000
	398.8889
	427.7778
	456.6667
	485.5556
	514.4445
	543.3333
	572.2222
	601.1111

Reach: R1 | Profile: 427.7778 | Timestamp : 24 days, 11:00:00 (2113200.0 sec)

Figure 10: Sedimentary results window for “*Elargissement*” tests case at timestamps 2113200.0 sec for reach R1 and cross-section at KP 427.7778. This example has only two sediment layers.

4. FUTURE WORK

Old PAMHyR functionalities and additional functionalities, allowed the definition of structures, traps, suspended solids or dissolved pollutants in water, have yet to be developed. Additional options for visualizing results also need to be added. One possibility would be a graph generation tool that would allow the user to select x-axis and y-axis values, as well as the time range, to create a customized graph. We also planned an advanced feature with code-editing window to enable the user to define his own Matplotlib drawing function, which would be saved in the study. Presentation work is also planned to harmonize the application's icons and visuals (images, graphics, etc.). The aim is to make the application clearer and more user-friendly.

Supporting new solvers and generic solvers will also be an important part of the remaining work. To use any solver, we need to define a generic output format for Pamhyr2 that is simple enough to be given as solver input without major code modification, or that can be processed by a simple Python script (for example). The same applies to the results file that Pamhyr2 will have to read. The main candidates for these generic formats are :

- SQLite, because it is easy to use and the library is compatible with most programming languages,
- HDF5[13], a hierarchical file format widely used in scientific circles and compatible with many programming languages.

As the project progresses, it becomes more and more complicated to test all the functionalities by hand. A suite of tests and unit tests is needed to prevent regression bugs and ensure that the code remains maintainable. For the moment, no such test has been defined. Unit tests will have to be developed on the model and each sub-module will have to include tests, but higher-level functionalities will also have to be tested, such as the recording and playback of a complete study. It will be difficult to test the graphical part automatically, so tests will continue to be carried out manually.

Full documentation must also be provided to give users an overview of the features offered by Pamhyr2. This documentation must include descriptions of the configuration options, as well as explanations and examples of how to use each feature. In addition, this documentation must include a section for contributors and developers, with a technical presentation of the application, as well as a presentation of the translation, development and debugging tools available.

For translations into other languages than English and French, we hope that external contributors will take on this work. But no languages other are currently planned by the development team. The application packages are currently available for GNU/Linux and Windows. Under GNU/Linux, a “.tar.xz” archive is available for download. For Windows, this is an installer for application and a compatible version of Mage. An installer for GNU/Linux systems is a work in progress. There are many choices for creating an installer, but their scope depends on the distribution for which they are intended. “.deb”, for example, is intended for Debian distributions and their derivatives (Ubuntu, Linux Mint, and so on). We do not exclude other, more generic options.

5. CONCLUSION

After a great deal of preparatory work, experimentation, technical selection, proof-of-concept and several months of development, Pamhyr2 is currently in version “v0.0.0”[14] and contains almost 20,000 lines of Python code. It allows to create a minimal 1D hydro-sedimentary study, run a Mage simulation and visualize some of the results obtained, all with a more user-friendly interface. But the software is still unstable and under development, and many modifications and improvements are still to be made before it can really be used by everyone. Furthermore, even though many of the features of the original version of PAMHyR are already available (see 3.1 page 3), there are still many to be added to make Pamhyr2 a complete, useful and fully functional piece of software.

[13] HDF5: https://www.hdfgroup.org/solutions/hdf5/
[14] Pamhyr2 releases: https://gitlab.irstea.fr/theophile.terraz/pamhyr/-/releases

ACKNOWLEDGEMENTS

The results presented here partially come from the Pamhyr PITI project funded by the *Carnot Eau & Environnement* institute and the French National Research Agency. *Carnot Eau & Environnement* was accredited in March 2020 by the French Ministry of Research and Higher Education on the proposal of the *Agence Nationale de la Recherche*. The signatory members are INRAE, CNRS, Aix Marseille University, IFTS, INSA Lyon, IRD, *Université de Montpellier* and *Université de Savoie Mont-Blanc*.

The authors thank Sylvain Coulibaly for his preparatory work and his technical choices.

REFERENCES AND CITATIONS

[1] Balayn, P. (2001). Contribution à la modélisation numérique de l'évolution morphologique des cours d'eau aménagés lors de crues. Université Claude Bernard - Lyon 1

[2] Camenen, B., Faure, J. Décanis, S. Diéval, L. (2021) A 1D numerical tool for real time modelling of a complex river network, SimHydro 2021

[3] Faure, J. (2013). An Integrated Simulation platform - PAMHyR. In John Wiley & Sons, Inc. eBooks (pp. 245–252). https://doi.org/10.1002/9781118557891.ch21

[4] Faure, J. (2017). Développement d'Adis-TS. OSR4 | Action V.1. https://hal.science/hal-03749339

[5] Faure, J-B., (2019). Mage, Résolution des équations de Barré de St Venant 1D en réseaux complexes, Documentation théorique et mode d'emploi. Techreport (in French)

[6] Walt, S. 2007. Free/open source software development. In Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (ESEC-FSE '07). Association for Computing Machinery, New York, NY, USA, 459–468. https://doi.org/10.1145/1287624.1287689